4. Chalam

Program Name: Chalam.java **Input File: chalam.dat**

Your friend Chalam has taken a recent fascination in the fundamental theorem of arithmetic which states that every positive integer can be represented uniquely as a product of prime numbers. Chalam's fascination has been in finding that unique prime representation for different numbers. However, he has recently come to a standstill as the numbers he wishes to break down are becoming increasingly larger and larger, making the amount of time that it takes to identify the unique classification unbearably long. Knowing a thing or two about how to automate this process, you decide to write Chalam a program which, given a positive integer, determines the unique prime product representation of that number.

Input: The first line of input will consist of a single integer, n ($1 \leq n \leq 2.5 \cdot 10^4$), denoting the number of integers that need to be processed. The next n lines will each consist of a single integer

q_i, (for all i, $1 \leq i \leq n$: $2 \leq q_i \leq 2^{31} - 1$),

the i^{th} of which represents Chalam's i^{th} query.

Output: For each of Chalam's n queries, each on their own line, output a space-separated list of the prime numbers, which, when multiplied with one another, equal Chalam's query. In cases where a given prime number is listed more than once, condense the expression into the form of $p_j^{e_j}$ (`p_j^e_j`) where p_j represents the j^{th} prime number and e_j represents the number of times that p_j occurs in the product. Lastly, these terms should be listed in ascending order according to the value of the base of the exponent (i.e., terms should be listed such that $p_1 < p_2 < \cdots < p_j$).

Sample input:

```
20
2
3
4
10
11
12
13
20
21
22
23
24
30
40
51
67
91
103
108
245
```

Sample output:

```
2
3
2^2
2 5
11
2^2 3
13
2^2 5
3 7
2 11
23
2^3 3
2 3 5
2^3 5
3 17
67
7 13
103
2^2 3^3
5 7^2
```